REMARKS

Claims 1-8 are pending in the application.

Double Patenting

Claims 1-8 have been rejected on the ground of non-statutory obviousness-type double patenting as being unpatentable over claims 1-5 of U.S. Patent No. 7,218,979.

A terminal disclaimer is submitted herewith to obviate the double-patenting rejection. Applicants, therefore, request that the double patenting rejection be withdrawn.

Claim Rejections - 35 U.S.C. § 112

Claims 2-8 have been rejected under 35 USC 112, second paragraph, as being indefinite. More specifically, the Examiner has stated that the term "interpreter-type" programming language is not clearly defined by the specification and is therefore indefinite.

In the present invention the variable program is created using an interpreter type programming language for CAD figure, called "Union Parts - BASIC". An interpreter type programming language is a programming language that is prepared to carry out while interpreting a written source code one by one. The program made from this program language does not require the compilation (compile means the computer directly converts the source code into the program of the machine language). The programs created using JAVA and the Common Object Request Broker Architecture (CORBA) need to be compiled. The advantage of not requiring to compile a variable program in the "Union Parts - BASIC" makes debugging and the updating of the variable program to be very simple. In addition, if the variable program is changed it can be accessed instantly by clients, with aid of network such as Internet. Further it is not necessary to reboot the server system.

When using the variable program by "Union Parts - BASIC" through the Internet it is not dependent on middleware such as JAVA Remote Method Invocation (RMI), HTTP Common

Gateway Interfaces(CGI), or CORBA ORBs (see Figure 3 of Saha). Therefore, the variable program can be detached from the system and saved in a database. This gives advantages because the variable program is not influenced when altering the system and vice versa. The possibility of storing the variable program in the database allows simple maintenance procedure. This is particularly important when tens of thousands of variable program units are managed.

JavaCadd used in Geppert, which is quoted by the examiner as an interpreter type programming language, is a program developed by program language called Java. Therefore, it is not a program language its self. In addition, since it is a program made by Java, it must be compiled with a java compiler. Further, it is dependent on RMI middleware of Java. This shows that it does not relate with "Union Parts - BASIC" which is an interpreter type programming language for CAD figure making "Union Parts Basics".

It is, therefore, believed that the term "interpreter-type" programming language is commonly understandable by a person skilled in the art and is thus not indefinite. Applicants, therefore, request that the rejection under 35 USC 112, second paragraph, be withdrawn.

Claim Rejections - 35 U.S.C. § 102

Claims 1-8 are rejected under 35 USC 102(b) as being anticipated by Saha et al. ("Web-Based Distributed VLSI Design," hereafter "Saha").

Claims 1-8 are rejected under 35 USC 102(b) as being anticipated by Geppert et al. ("IC Design on the World Wide Web," hereafter "Geppert").

Saha concerns a Web-based distributed VLSI (Very Large Scale Integration) design, which is the process of creating integrated circuits by combining thousands of transistor-based circuits into a simple chip. VLSI system requires distributed design and verification methodology due to the diverse expertise required at various areas. The system requires tools and information which is accessible through the Internet. Figure 2 of Saha shows the use of the Web to identify various source of information which mostly deal with CAD environment to estimate power dissipation. This is clearly different from the present invention which uses the Internet to

send the information to construct **parametric** graphic data. The PowerZone, which is made by designers using Web tools, is used to estimate the power loss, but does not specify the use of parametric graphic data to create the display data. In addition, the power play tool is also available in the Web as a tool to facilitate engineers to verify the power consumption. The seamless access to cell libraries and VLSI CAD tools such as synthesizers, generators, optimizers and simulators are clearly the ways to facilitate engineers. In contrast, in the present invention the clients only access the Internet so the required data are sent to the clients to construct the **parametric** graphical data and parts model number.

Geppert describes an IC Design method in which the parts are searched using many methods such as search engines. The parts found are for ordering. In addition it has the options to allow checking for availabilities. This clearly shows that the parts stored are found in order to place orders. Again, in contrast to Geppert, in the system of the present invention the client computer chooses its variable program and its numerical data from the list to make the most suitable parametric graphic data then converts into display data to be integrated into their CAD drafts and not to place orders.

In the case of IC design, information is allowed to be passed though Internet to ease engineers in finding information. However in the case of the present invention, the only information passed through the Internet to the client computer is the raw data. This concludes that the data passed with aid of the network in the present invention is not related at all to the data passed through Internet in IC design. In IC design, there is a system that allows collaborative design on the Web so that the team can access from anywhere around the world and allows the designers to check on the database and transmit over the Web. This method is there to allow the team to access its information from anywhere but does not show the use of that information to construct any graphical data in the manner as constructed in the present invention.

The Geppert's "Part Numbers" and Saha's parts/model numbers that the examiner quoted are the part/model numbers prepared beforehand. These parts/model numbers are not automatically generated according to the client's request, and therefore they can only be used for the parts/model numbers prepared beforehand. In contrast, the present invention does not have

to prepare the part numbers before hand, and according to the client's request the numerical data prepared beforehand, or else inputted by the client, are substituted into the variable program to generate the graphic (figure) data and part model numbers. This is called the parametric method. By inputting the numerical data, it provides the possibilities to generate infinite number of graphic data and parts model numbers. For example: a part model number changes according to the bolt length. In the present invention, the bolt lengths are substituted in the variable program for generating the parametric graphic data according to the substituted data, and for further generating the part model number. The numerical data and variable data are stored in the server. According to the client's request, the particular numerical data and variable data are forwarded to the client through the network connection such as the Internet. These data are used in the client computer to generate the parametric graphic data and the parts model number. The data are kept in the server computer. Keeping the data in the server gives considerable amount of merit such as updating, withdrawing and adding new data.

In addition, the graph plotting in Saha's "graph plotting utility" and Web based CAD only calls the XY coordinates from the file and plots the graph figures according to its file. In contrast, in the present invention, according to the client's request the numerical data prepared beforehand, or else inputted by the client, are substituted into the variable program to generate the graphic (figure) data and the part model numbers. This is called the parametric method. This parametric method uses a variable program for a shape and many different sizes can be generated endlessly. For example: a bolt has its specific shape with many different sizes available. The specifications are met by using one variable program for its shape and "many" numerical data that corresponds to its magnitude.

The ability for users to create the companies web page to report the result of power analysis in the cited reference is again used so the team can access the web page to gather information. Again no such system exists in the present invention where web is just used to allow clients to request its specification so its correspondent parts are displayed in their display device. The basic data which is received in the data receiving section is just there to receive the raw data according to the client's request. These raw data is only used to create parametric

graphic data. The data are not used for clients to find information such as power dissipation but to create display data (different purposes).

Finally, as explained in connection with the rejection under 35 USC 112, second paragraph, none of the cited references uses an interpreter type programming language.

The present invention as claimed is, therefore, believed to be patentable over the art and the Examiner is requested to withdraw the rejections under 35 USC 102 .

Favorable consideration and early issuance of the Notice of Allowance are respectfully requested. **Should further issues remain prior to allowance, the Examiner is respectfully requested to contact the undersigned at the indicated telephone number.**

Respectfully submitted,

Date: February 29, 2008

Yonghong Chen (Registration No. 56,150)
Akerman Senterfitt
Customer No. 30448
222 Lakeview Avenue, Suite 400
West Palm Beach, FL 33401
Phone: 561-653-5000